

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ**

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_  
(підпис)      Тарасенко В.П.  
(ініціали, прізвище)

“ \_\_\_\_ ” червня \_\_\_\_ 2019 р.

**Дипломний проект  
на здобуття ступеня бакалавра**

з напрямку підготовки      6.050102 «Комп'ютерна інженерія»

на тему: Веб-орієнтований додаток збору статистичної інформації для відеострімінгового сервісу

Виконав: студент IV курсу, групи КВ-53

Яковенко Олександр Геннадійович

— \_\_\_\_\_  
(підпис)

Керівник доцент кафедри СПіСКС,

к.т.н. О.А. Щербина

(підпис)

Консультант з нормоконтролю к.т.н. Клятченко Я.М.

— \_\_\_\_\_  
(підпис)

Рецензент

\_\_\_\_\_

(підпис)

Засвідчую, що у цьому дипломному  
проекті немає запозичень з праць інших  
авторів без відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2019 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки 6.050102 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Тарасенко В.П.  
(підпис) (ініціали, прізвище)

«\_\_\_» \_\_\_\_\_ червня \_\_\_\_\_ 2019 р.

**ЗАВДАННЯ**  
**на дипломний проект студента**  
**Яковенка Олександра Геннадійовича**  
(прізвище, ім'я, по батькові)

1. Тема проекту: Веб-орієнтований додаток збору статистичної інформації для відеострімінгового сервісу,  
керівник проекту доцент кафедри СПіСКС, к.т.н. Щербина О.А.,  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «25» травня 2019 р. №1330-С

2. Термін подання студентом проекту: «10» червня 2019 р.

3. Вихідні дані до проекту: науково-технічна література щодо веб-додатків та прикладних програмних програмних інтерфейсів.

4. Зміст пояснювальної записки: аналіз існуючих рішень та обґрунтування теми дипломного проекту; обґрунтування вибору засобів реалізації; структурно-алгоритмічна організація; опис програмного продукту; висновки.

5. Перелік графічного матеріалу:

ІАЛЦ.045440.005 Д1 Модульна структура додатку;

ІАЛЦ.045440.006 Д2 Асинхронна модель додатку;

ІАЛЦ.045440.007 Д3 Архітектура додатку;

ІАЛЦ.045440.008 Д4 Алгоритм роботи розробленого додатку.

6. Консультанти розділів проекту<sup>†</sup>

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М.		

7. Дата видачі завдання \_\_\_\_\_

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1	Видача завдання на дипломне проектування	16.12.2018	
2	Розробка технічного завдання	07.02.2019	
3	Аналіз існуючих рішень	12.03.2019	
4	Розробка програмного продукту	23.03.2019	
5	Відлагодження програмного продукту	30.03.2019	
6	Підготовка пояснювальної записки	14.05.2019	
7	Оформлення матеріалів проекту	19.05.2019	
8	Попередній огляд матеріалів диплому на кафедрі	23.05.2019	

Студент

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
(ініціали, прізвище)

Керівник проекту

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
(ініціали, прізвище)

<sup>†</sup> Консультантом не може бути зазначено керівника дипломного проекту.

## Зміст

1	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ	2
2	ПІДСТАВА ДЛЯ РОЗРОБКИ	2
3	МЕТА І ПРИЗНАЧЕННЯ РОБОТИ	2
4	ДЖЕРЕЛА РОЗРОБКИ	2
5	ТЕХНІЧНІ ВИМОГИ	2
5.1	Вимоги до програмного продукту, що розробляється . . . . .	2
5.2	Вимоги до апаратного забезпечення . . . . .	3
5.3	Вимоги до програмного та апаратного забезпечення користувача .	3
6	ЕТАПИ РОЗРОБКИ	4

					ІАЛЦ. 045440.002 ТЗ			
Зм.	Лист	№ докум.	Підп.	Дата				
Розробив	ЯковенкоО.Г.				Веб-орієнтований додаток збору статистичної інформації для відеострімінгового сервісу Технічне завдання	Лит.	Аркуш	Аркушів
Перевірів	ЩербинаО.А.						1	4
						КПІ ім. Ігоря Сікорського, ФПМ, КВ-53		
Н. контр.	КлятченкоЯ.М.							
Затвердив	ТарасенкоВ.П.							

## 1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Веб-орієнтований додаток збору статистичної інформації для відеострімінгового сервісу».

Галузь застосування: медіа-сфера, кібер-спорт.

## 2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання дипломного проекту першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

## 3 МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є створення платформи для стрімінгу та збір статистики з віддалених сервісів.

## 4 ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

## 5 ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до програмного продукту, що розробляється

- Сумісність з операційними системами Windows, Linux, MacOS.
- Можливість отримання статистики.
- Можливість відображення трансляції.

					ІАЛЦ. 045440.002 ТЗ	Аркуп
Зм.	Лист	№ докум.	Підп.	Дата		2

- Можливість відправлення повідомлень та перегляду чату каналу.
- Можливість авторизації.
- Можливість збереження даних про стрімера в БД.
- Можливість зміна даних.

## 5.2 Вимоги до апаратного забезпечення

- Процесор: 1,2,4-ядерний, Intel, AMD, ARM.
- Оперативна пам'ять: 4 Гб.

## 5.3 Вимоги до програмного та апаратного забезпечення користувача

- Інтерпретатор мови JavaScript.
- Бібліотеки для мови JavaScript: React.

					ІАЛЦ. 045440.002 ТЗ	Аркуп
Зм.	Лист	№ докум.	Підп.	Дата		3

## 6 ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів
1.	Вивчення літератури за тематикою проекту	18.12.2018
2.	Розроблення та узгодження технічного завдання	28.01.2019
3.	Аналіз існуючих рішень	06.02.2019
4.	Підготовка матеріалів першого розділу дипломного проекту	11.04.2019
5.	Підготовка матеріалів другого розділу дипломного проекту	17.05.2019
6.	Підготовка графічної частини дипломного проекту	20.05.2019
7.	Оформлення документації дипломного проекту	27.05.2019
8.	Попередній огляд матеріалів диплому на кафедрі	28.05.2019

## АНОТАЦІЯ

Кваліфікаційна робота містить пояснювальну записку (50 сторінка, 22 рисунки, 3 таблиць).

Дипломний проект містить односторінковий веб-додаток, що є розважальною платформою, в основі якої кібер-спортивні дисципліни. Основною метою проекту являється демонстрація використання API у веб-додатках на конкретному прикладі.

Даний додаток створений для використання усіма верствами населення, які бажають ділитись своїми діями, захопленнями, творіннями та іншим. Передбачена можливість створювати власний інтерфейс, авторизація, ведення трансляцій, перегляд статистики, а також грошові транзакції.

Ключові слова:

ВЕБ-ДОДАТОК, JAVASCRIPT, REACT, GraphQL, API, GITHUB, ВЕБ-ІНТЕРФЕЙС, БІБЛІОТЕКА, PostgresQL.



## ABSTRACT

The qualifying paper contains an explanatory note (page-50, drawings-22, tables-3).

The diploma project contains a one-page web application that is an entertaining platform based on cyber-sport disciplines. The main purpose of the project is to demonstrate the use of the API in web applications for a specific example.

This application is designed for use by all strata of the population, who want to share their actions, hobbies, creatures and others. Provides the ability to create your own interface, authorization, maintenance broadcasts, statistics, as well as cash transactions.

Keywords:

WEB APPLICATION, JAVASCRIPT, REACT, GraphQL, API, GITHUB, WEB INTERFACE, LIBRARY, PostgresQL.

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кіл-ть. арк.	№ прим.	Прим.
			<u>Документація загальна</u>			
			<u>Новорозроблена</u>			
	A4	IАЛЦ.045440.002 ТЗ	Веб-орієнтований додаток збору статистичної інформації для відеострімнігового сервісу Технічне завдання	4		
	A4	IАЛЦ.045440.003 ТП	Веб-орієнтований додаток збору статистичної інформації для відеострімнігового сервісу Відомість технічного проекту.	2		
	A4	IАЛЦ.045440.004 ПЗ	Веб-орієнтований додаток збору статистичної інформації для відеострімнігового сервісу Пояснювальна записка	55		
	A1	IАЛЦ.045440.005 ДІ	Модульна структура веб-дodatku. Схema структурна	1		
Zмін.	Арк.	№ докyм.	Підпис	Дата	IАЛЦ.045440.001 OА	
Rозробив	Яковенко О.Г.				Lіт.	Аркyш Аркyшів
Pеревір.	Щербина О.A.					1 2
H. контроль	Kлятченко Я.M.				KPІ ім. Iгоря Cікорського, ФПM, KB-	
Zатвердив	Tарасенко B.П.				53	
						Oпис альбомy

[illegible]

## Зміст

<b>1</b>	<b>НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ</b>	<b>2</b>
<b>2</b>	<b>ПІДСТАВА ДЛЯ РОЗРОБКИ</b>	<b>2</b>
<b>3</b>	<b>МЕТА І ПРИЗНАЧЕННЯ РОБОТИ</b>	<b>2</b>
<b>4</b>	<b>ДЖЕРЕЛА РОЗРОБКИ</b>	<b>2</b>
<b>5</b>	<b>ТЕХНІЧНІ ВИМОГИ</b>	<b>2</b>
5.1	Вимоги до програмного продукту, що розробляється . . . . .	2
5.2	Вимоги до апаратного забезпечення . . . . .	3
5.3	Вимоги до програмного та апаратного забезпечення користувача .	3
<b>6</b>	<b>ЕТАПИ РОЗРОБКИ</b>	<b>4</b>

					ІАЛЦ. 045440.002 ТЗ							
Зм.	Лист	№ докум.	Підп.	Дата	Веб-орієнтований додаток збору статистичної інформації для відеострімінгового сервісу Технічне завдання				Лит.	Аркуш	Аркушів	
Розробив	ЯковенкоО.Г.										1	4
Перевірив												
Н. контр.	КлятченкоЯ.М.											
Затвердив									НТУУ «КПІ ім. Ігоря Сікорського», ФПМ, КВ-53			

## 1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Веб-орієнтований додаток збору статистичної інформації для відеострімінгового сервісу».

Галузь застосування: медіа-сфера, кібер-спорт.

## 2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання дипломного проекту першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

## 3 МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є створення платформи для стрімінгу та збір статистики з віддалених сервісів.

## 4 ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

## 5 ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до програмного продукту, що розробляється

- Сумісність з операційними системами Windows, Linux, MacOS.
- Можливість отримання статистики.
- Можливість відображення трансляції.

					ІАЛЦ. 045440.002 ТЗ	Аркул
Зм.	Лист	№ докум.	Підп.	Дата		2

- Можливість відправлення повідомлень та перегляду чату каналу.
- Можливість авторизації.
- Можливість збереження даних про стрімера в БД.
- Можливість зміна даних.

## 5.2 Вимоги до апаратного забезпечення

- Процесор: 1,2,4-ядерний, Intel, AMD, ARM.
- Оперативна пам'ять: 4 Гб.

## 5.3 Вимоги до програмного та апаратного забезпечення користувача

- Інтерпретатор мови JavaScript.
- Бібліотеки для мови JavaScript: React.

					ІАЛЦ. 045440.002 ТЗ	Аркул
						3
Зм.	Лист	№ докум.	Підп.	Дата		

## 6 ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів
1.	Вивчення літератури за тематикою проекту	15.04.2019
2.	Розроблення та узгодження технічного завдання	30.04.2019
3.	Аналіз існуючих рішень	05.05.2019
4.	Підготовка матеріалів першого розділу дипломного проекту	10.05.2019
5.	Підготовка матеріалів другого розділу дипломного проекту	18.05.2019
6.	Підготовка графічної частини дипломного проекту	20.05.2019
7.	Оформлення документації дипломного проекту	25.05.2019
8.	Попередній огляд матеріалів диплому на кафедрі	30.05.2019

[illegible]



[illegible]

## ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ	3
ВСТУП	5
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ	6
1.1 Обґрунтування теми	6
1.2 Аналіз існуючих додатків	7
2. ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ	11
2.1 Вибір мови програмування для розробки	11
2.2 Обґрунтування вибору технологій для розробки	16
2.2.1 Обмін даними між клієнтом та сервером	16
2.2.2 Вибір формату даних для обміну	17
2.2.3 Вибір фреймворку для написання логіки	19
2.2.4 Вибір фреймворку для написання стилів	21
2.2.5 Вибір баз даних	23
2.3 Огляд обраних API для використання у розробці	25
2.4 Вибір засобів розробки	26
3. СТРУКТУРНО-АЛГОРИТМІЧНА ОРГАНІЗАЦІЯ	30
3.1 Загальний опис структури та роботи файлів проекту	30
4. ОПИС ПРОГРАМНОГО ПРОДУКТУ	33
4.1 Принцип роботи додатку	33
ВИСНОВКИ	47
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	48

					ІАЛЦ.045440.004 ПЗ			
Змін	Арк.	№ докум.	Підпис	Дата	Веб-орієнтований додаток збору статистичної інформації для відеострімінгового сервісу  Пояснювальна записка	Літ.	Аркуш	Аркушів
Розроб.		Яковенко О.Г.					1	50
Перевір.		Орлова М.М.						
Н. контр.		Клятченко Я.М.				КПІ ім. Ігоря Сікорського, ФПМ, КВ-53		
Затвер.		Тарасенко В.П.						

## Додатки

### Додаток 1. Копії графічних матеріалів

ІАЛЦ.045440.005 Д1 Класична та асинхронна моделі веб-додатку.

ІАЛЦ.045440.006 Д2 Архітектура React додатку.

ІАЛЦ.045440.007 Д3 Схема викликів функцій розробленого додатку.

ІАЛЦ.045440.008 Д4 Алгоритм роботи розробленого додатку.

### Додаток 2. Фрагменти програмного коду

### Додаток 3. Презентація

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		2

## СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

**CMS** (англ. Content Management System) – програмне забезпечення для організації веб-сайтів чи інших інформаційних ресурсів в Інтернеті чи окремих комп'ютерних мережах.

**ERP** (англ. Enterprise Resource Planning System) – корпоративна інформаційна система, призначення для автоматизації обліку й управління.

**HSL** (англ. Hue, Saturation, Lightness) – колірна модель, в якій будь-який колір визначається трьома характеристиками: кольоровим тоном, насиченістю та освітленістю.

**XML** (англ. Extensible Markup Language) – стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками, зокрема через Інтернет.

**RGB** (англ. Red, Green, Blue) – адаптивна колірна модель, що описує спосіб синтезу кольору, за якою червоне, зелене та синє світло накладаються разом, змішуючись у різноманітні кольори.

**RGBA** (англ. Red, Green, Blue, Alpha) – аналогічна до RGB колірна модель, але з додаванням коефіцієнту прозорості.

**XHTML** (англ. Extensible Hypertext Markup Language) – мова розмітки, що має таку саму виразну силу як і HTML але відповідає синтаксичним правилам XML.

**SVG** (англ. Scalable Vector Graphics) – специфікація мови розмітки, що базується на XML, та формат файлів для двовимірної векторної графіки, як статичної, так і анімованої та інтерактивної.

**CRM** (англ. Customer relationship management) – поняття, що охоплює концепції, котрі використовуються компаніями для управління взаємовідносинами зі споживачами, включаючи збір, зберігання й аналіз інформації про споживачів, постачальників, партнерів та інформації про взаємовідносини з ними.

**Рефактор**(англ. refactor) – перетворення кода на більш за зрозумілий та оптимізований код.

**Колбек** (англ. callback) – функція зворотного виклику, частина виконуваного коду, що передається як аргумент до іншого коду, який має викликати цей код у відповідь, тобто виконати аргумент у певний момент часу.

**Байндинг** (англ. binding) – зв’язування даних, процес, що встановлює зв’язання між користувацьким інтерфейсом додатку та бізнес-логікою.

**Сайдбар** - бокова панель.

**JSON** (англ. JavaScript Object Notation) – текстовий формат обміну даними між комп’ютерами. Дозволяє описувати об’єкти та інші структури даних.

**SSH** (англ. Secure Shell) – мережевий протокол рівня застосунків, що дозволяє проводити віддалене управління комп’ютером і тунелювання TCP-з’єднань (наприклад, для передачі файлів).

**Нативний** (лат. nativus) – не модифікований.

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		4

## ВСТУП

В сучасному світі дуже стрімко розвивається сфера онлайн ігор, можна відмітити, що дана сфера захопила дуже велику частку населення. Стрімер - одна з найпопулярніших та найдоступніших, так би мовити, професій, пов'язаних з комп'ютерними іграми. Найпопулярніша платформа для стрімерів - Twitch. Дана платформа надає можливість отримувати дані стосовно кожного каналу за допомогою API.

В даному дипломному проєкті створено веб-додаток, який дозволяє кожному отримати та розглянути детальну статистику власного каналу.

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		5

# 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

## 1.1 Обґрунтування теми.

Дана тема дипломного проекту була обрана з метою показати, що за допомогою вже існуючих API, можна легко створити власну міні-платформу і наскільки корисним та доцільним є використання вже існуючих додатків.

Загалом спираючись на сучасний світ, де гаджети є частиною життя майже кожної людини, а також дуже стрімкий розвиток ігор та кількість людей, які намагаються свої захоплення, перетворити на власну справу, методом розповсюдження своєї діяльності в онлайн ефірах, то створення подібної платформи, з простим у використанні інтерфейсом, як для самого стрімера, так і для користувача, є перспективним.

## 1.2 Аналіз існуючих додатків.

Проаналізувавши вже існуючі подібні рішення, та створивши, платформу не гіршої якості з своїми віджетами. Найпопулярніші платформи: Twitch.tv[31], YouTube.com[32], CyberGame.tv[33].

### 1. Стрім на Twitch.tv

Міжнародна платформа для стріму, існуюча за мірками інтернету вже дуже давно - з 2011 року. Головною її перевагою є величезна аудиторія - сайт відомий по всьому світу і транслює багато популярних турнірів. Він найбільш розкручений на цьому ринку, але і конкуренція на ньому набагато вище.

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		6

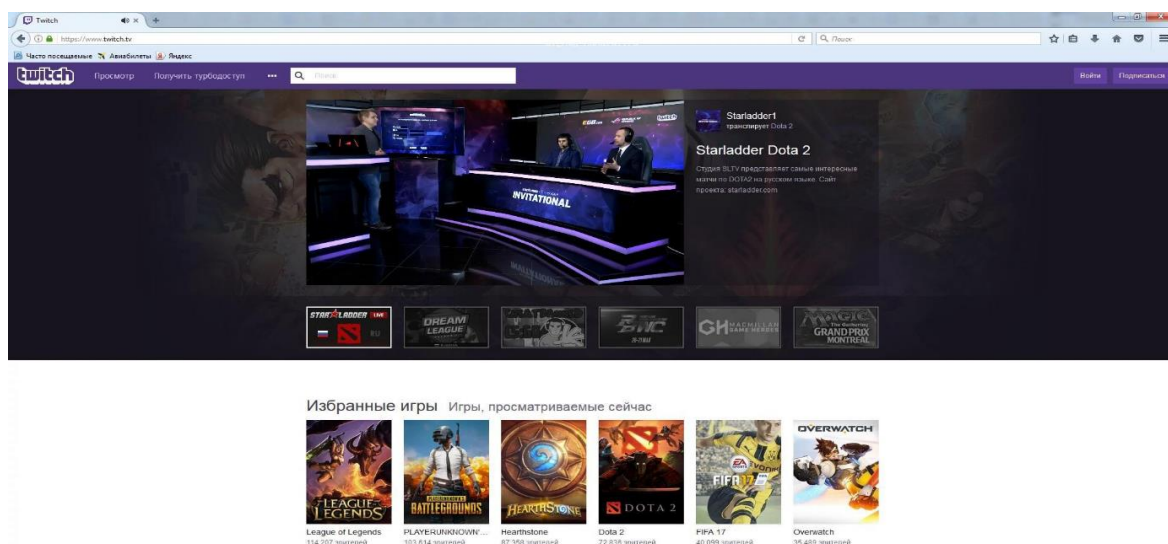


Рисунок 1.1 – Скріншот сайту twitch.tv.

Ресурс також профілюється на іграх. Традиційно вважається, що саме Twitch найкращий в своєму роді, а конкуренти ще не доросли до його рівня. При цьому і на Twitch, хоч і зрідка бувають перебої з відео. До плюсів же можна віднести:

- Найбільшу аудиторію.
- Чіткі і докладні умови співпраці.
- Дотримання закону про авторські права.
- Якість і стабільність плеєра.
- Практично всі значущі розробники ігор, ЗМІ про них, мають тут свої потоки, демонструють з'їзди і конференції.
- Існує спеціальний розділ «креатив», в якому можна транслювати етапи своєї творчості.
- Популярність сайту в світі говорить сама за себе - більшість стримеров досягли популярності саме на twitch.



## 2. Трансляція на YouTube.com

YouTube, як один з найпопулярніших сайтів в світі в цілому вже давно дістався і до трансляції стрімів. Маючи один з кращих і стабільних плеєрів в інтернеті, йому не потрібна була навіть особлива реклама - аудиторія прийшла сама.

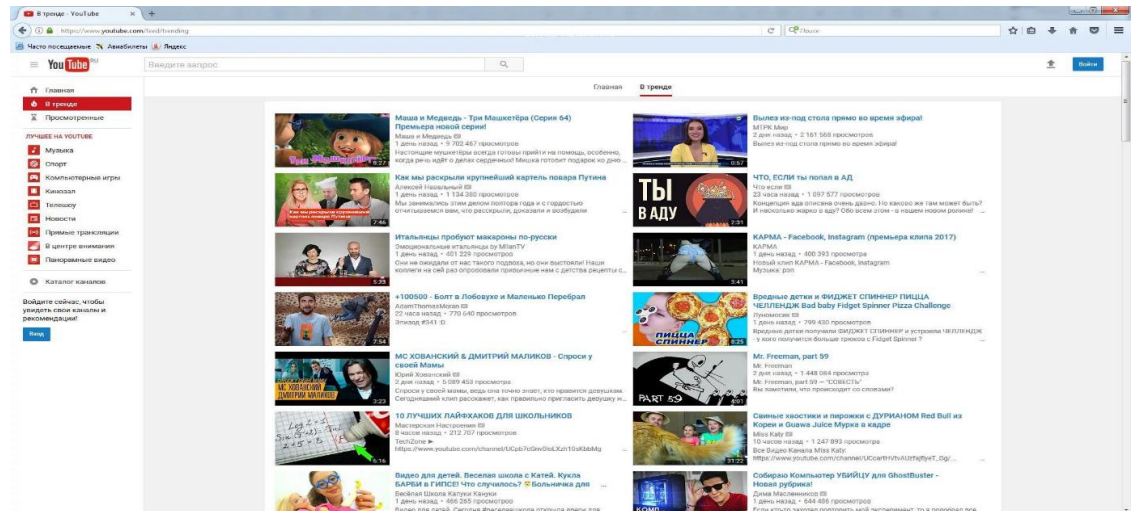


Рисунок 1.2 – Скріншот сайту youtube.com.

### Переваги youtube:

- Велика і різношерста аудиторія, з якою можна залучити нових глядачів.
- Можна проводити стрім на абсолютно будь-які теми - тільки б зацікавити оточуючих.
- Відмінний плеєр і функціонал.
- При цьому досить велика конкуренція, при якій домогтися популярності буде важкувато. Ідеально, якщо youtube стане другим каналом, на який буде йти потік, особливо якщо мова про ігровий тематиці.

### 3. Трансляція на CyberGame.tv

Ще одним відомим сайтом для стріму в ігровій тематиці є CyberGame. Досить велика аудиторія і тлумачний підхід розробників дозволяють йому стабільно утримуватися на плаву.

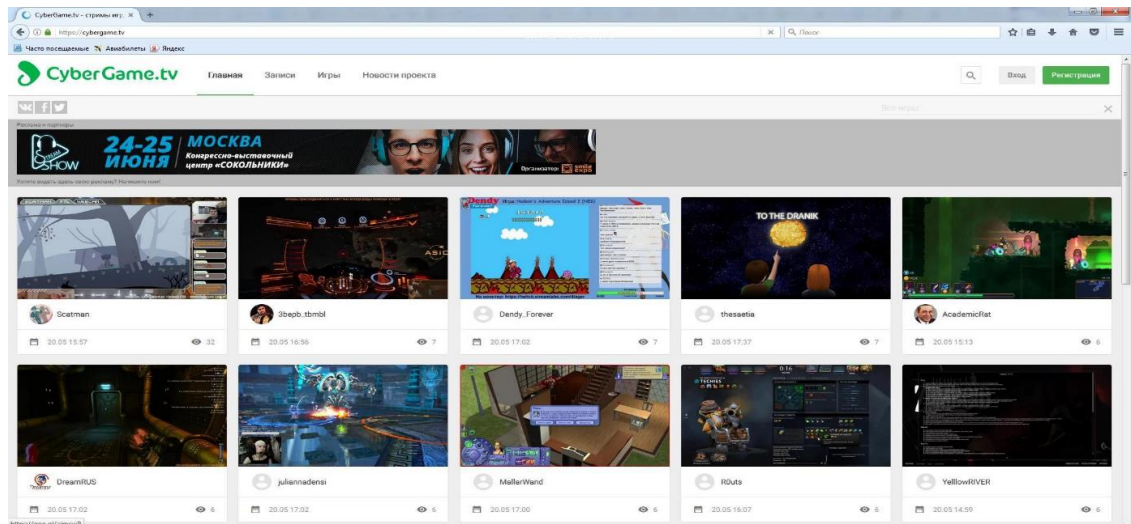


Рисунок 1.3 – Скріншот сайту CyberGame.tv.

Серед переваг варто відзначити:

- Постійні оновлення функціоналу, додавання нових фішок;
- Хороший, настроюється, працює плеєр, який дозволяє залучати, а не відштовхувати аудиторію;
- Наявність преміум акаунтів;
- Сайт сам по собі зручний для глядача, є багато розділів по найпопулярнішим іграм, що дозволяє аудиторії швидше «прийти до свого» стримеру;
- Трансляція декількох фестивалів і турнірів;
- Зручний чат з безліччю смайлів.

Отже, проаналізувавши найуспішніші платформи, можна зробити висновок, що всі вони охоплюють дуже великий спектр користувачів, іноді є недотепним, адже велика кількість функціоналу залишається не

використаною, тому створено нову вузькоспеціалізовану платформу, яка не містить нічого зайвого, та завдяки віджетам дозволяє виконувати дії ніколи не відволікаючись від трансляції.

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		10

## 2. ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ

### 2.1 Вибір мови програмування для розробки.

Основними критеріями для вибору мови були: можливість взаємодіяти з зовнішніми API, кросплатформність та створити власний користувацький інтерфейс. Веб-сторінки, як правило, створюються за допомогою мови HTML (з англ. - Мова розмітки гіпертекстових документів).

Для даного проекту було обрано HTML5 – останню версію мови HTML. Остання версія містить семантичні теги основних елементів: <header> (заголовок розділу), <footer> (визначає нижню частину розділу), <nav> (навігація по сайту), <aside> (вміст спорідненого основному контенту), <section> (представляє основний контент документа), <article> (незалежний розділ документа), та інші. Семантичні теги створені для покращення структури сторінки, додавши смислове значення укладеному в них змісту. Пошукові системи, збирачі інформації та мовні браузерери краще аналізують веб-сторінки написані семантичними тегами.

Можна виділити ще такі нові цікаві теги, як **datagrid** (таблиця, яка побудована через динамічне джерело даних), **menu** (система меню), **command** (певні дії при активації об'єкта).

До існуючих типів полів HTML-форм додано нові: datetime, datetime-local, date, month, week, time, number, range, email, url.

Необхідно розуміти, що впровадження стандартів HTML5 не може відбутися миттєво, оскільки потрібен час на те, щоб всі браузерери впровадили їх у своїх нових версіях, а розробники – у свою практику. Поки що доступні хоч і багато, але не всі функції HTML5, і в різних браузерах їх набір відрізняється. Але уже зараз очевидно, що це – справа дуже

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		11

близького майбутнього, тому розробка сучасного сайту обов'язково включає стандарти HTML5.

Для створення зовнішнього вигляду веб-сторінки використовується CSS (з англ. Каскадні таблиці стилів) - мова, що використовується для візуальної презентації сторінок, написаних на HTML та XHTML.

CSS використовується розробниками веб-сторінок для завдання кольорів, шрифтів, розташування і інших аспектів представлення документа. Основною метою розробки CSS було розділення вмісту (написаного на HTML або іншій мові розмітки) від представлення стилю документа. Це розділення може збільшити доступність документа, надати велику гнучкість і можливість управління його виглядом, а також зменшити складність і повторюваність в структурному вмісті. Крім того, CSS дозволяє представляти один і той же документ в різних стилях.

Стандарт CSS визначає пріоритети, у порядку яких застосовуються правила стилів, якщо для якогось елемента підходять деякі правила одночасно. Це називається "каскадом", в якому для правил розраховуються пріоритети або "ваги", що робить результати передбаченими.

Таблиця стилів складається з набору правил. Кожне правило, у свою чергу, складається з одного або декількох селекторів, розділених комами і блоку визначень.

До появи CSS оформлення веб-сторінок здійснювалося безпосередньо усередині вмісту документа. Проте з появою CSS стало можливим принципове розділення змісту і представлення документа. За рахунок цього нововведення стало можливим легке застосування єдиного стилю оформлення для маси схожих документів, а також швидка зміна цього оформлення.

Переваги CSS розмітки:

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		12

- Декілька дизайнів сторінки для різних пристроїв перегляду. Наприклад, на екрані дизайн буде розрахований на велику ширину, під час друку меню не виводитиметься, а на КПК і стільниковому телефоні меню буде внизу за вмістом.
- Зменшення часу завантаження сторінок сайту за рахунок перенесення правил представлення даних в окремий CSS-файл. В цьому випадку браузер завантажує тільки структуру документа і дані, що зберігаються на сторінці, а представлення цих даних завантажується браузером тільки один раз і кешується.
- Простота подальшої зміни дизайну. Не потрібно правити кожен сторінку, а лише змінити CSS-файл.
- Додаткові можливості оформлення. Наприклад, за допомогою CSS-розмітки можна зробити так, щоб меню було завжди видно при скролінгу сторінки.

В даному проекті використовується остання версія CSS3. Перевага CSS3 - збільшена швидкість завантаження веб-сторінок, пов'язана зі зменшенням розмірів файлів .css. Створені нові функції, які збільшують можливості мови та полегшують створення зовнішнього вигляду сторінки: межі (кольору, зображення, радіусу), фони, кольори (HSL, HSLA, RGBA), текстові ефекти (текст-тінь, текст-переповнення, перенос слів), користувацький інтерфейс (коробка-розмірів, розмір, контур), селектори атрибутів, переповнення (x, y), CSS3 переходи, веб-шрифти, та інше. Також при створенні сторінки враховувалось адаптивність під пристрої з різним екраном.

Для зручності написання коду була обрана SASS – скриптова метамова, яка інтерпретується в каскадні таблиці стилів (CSS). Існують

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		13

такі синтаксиси даної метамови: SASS (синтаксис реалізований за допомогою табуляції) та SCSS (новий синтаксис; фігурні дужки присутні). Для даної роботи був використаний новий синтаксис метамови.

Скриптові мови програмування, які використовуються для написання логіки клієнтської сторони транспілюються в JavaScript - єдина мова програмування, що підтримується браузерами. Іноді доцільним є використання іншої мови з своїми особливостями.

JavaScript – динамічна, об'єктно-орієнтована мова програмування. В проєкті використовується стандарт EcmaScript6 з новими можливостями(деструктуризація, стрілкові функції, шаблонні рядки і т.д.)

EcmaScript 6 - нова версія популярного мови програмування javascript і напевно Ви вже про неї чули. В даному матеріалі розберемося, які корисні можливості представив новий реліз.

EcmaScript - мова, яка розвивається динамічно, тому його еволюція в технічному плані безперервно триває. EcmaScript відповідає за синтаксис мови, типи, прототипи і спадкування, стандартну бібліотеку вбудованих об'єктів і функцій, а саме: JSON, Math, методи масивів, методи об'єктів і інше. Крім того покриває ті аспекти мови, які представлені не тільки в веб-браузерах, але і в платформах типу node.js.

За результатами нового шостого релізу можна виділити головні наступні нововведення і властивості:

1. Деструктивне привласнення або деструктуризація (Destructuring assignment).
2. Деструктуризація аргументів у функціях.
3. Функція () => (стрілочки функції).
4. Шаблонні рядки (Template strings).
5. Оголошення змінних за допомогою let.

6. З'явилися класи, які спрощує розуміння не всім зрозумілого прототипного спадкування.

7. Додано безліч методів для рядків, масивів і математичних операцій.

Багато з цих властивостей вже працюють в сучасних і популярних веб-браузерах: Chrome і Firefox.

Модульна система ECMAScript 6 досить схожа на модулі відомої Node.js. Модуль - файл з javascript кодом всередині

Для роботи з EcmaScript 6 можна успішно використовувати транскомпілятор Babel. Він переводить код з нової версії в стару, більш стабільну.

І це відмінний варіант, так як можна протестувати можливості технології не встановлюючи її. А в разі потреби написання чогось більш серйозного зробити установку з сайту . Застосовуючи Babel інструмент код, написаний на ES6, легко компілюється в "класичний" стандарт ES5, в результаті веб-браузери з легкістю його сприймають.

Новий ECMAScript 6 має досить багато відмінностей від попередніх версій мови. Ці нововведення результат довгої та кропіткої роботи фахівців в даній галузі. Саме вони створили багато нових і корисних можливостей ЕСМА. Можна стверджувати, що це найважливіше оновлення JS, яке було з початку його появи. Створені функції починаються від спрощених зручностей на кшталт функцій-стрілок і інтерполяції рядків до серйозних концепцій на зразок проксі і генераторів.

Мета - шляхом розгляду нових особливостей показати розробнику, як ECMAScript 6 радикально і повністю поміняє Ваш код.

Як висновок можна тільки сказати, що ECMAScript 6 - це досить цікавий і корисний мову з великими можливостями javascript, який в найближчому майбутньому знайде популярність в сфері розробки.

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		15



## 2.2 Обґрунтування вибору технологій для розробки

### 2.2.1 Обмін даними між клієнтом та сервером

Розроблений веб-додаток буде виконувати запити до сторонніх API сервісів в якості серверів, тому треба обрати стек технологій та типи даних, для використання.

Між синхронною і асинхронною моделлю є суттєва різниця.

Синхронна модель:

- активація певного елементу веб-сторінки;
- браузер формує і відправляє запит на сервер;
- сторінка блокує можливість виконання дій;
- на відповідь від серверу отримується нова згенерована сторінка, перезавантажується вкладка та рендериться нова сторінка.

Асинхронна модель:

- активація певного елементу веб-сторінки;
- визначається інформація необхідна для оновлення сторінки за допомогою скрипта ;
- браузер виконує запит на сервер;
- сервер повертає лише зміни необхідної частини сторінки;
- за допомогою скрипта оновлюється сторінка відповідно до отриманої інформації(без повного перезавантаження сторінки).

З аналізу цих моделей, можна зробити висновок, що асинхронна модель більш доцільна для використання.

GraphQL - це синтаксис, який описує як запросити дані і, в основному, використовується клієнтською стороною для загрузки даних з серверу. Дана технологія має три основні характеристики:

- дозволяє клієнтській стороні точно вказати, які дані потрібні;

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		16

- полегшує агрегацію даних з декількох джерел;
- використовує систему типів для опису даних;

GraphQL – асинхронний, тому користувач може повноцінно користуватись сторінкою, поки сервер ще обробляє запит.

GraphQL навідмінну від REST набагато комфортніший в використанні, тому що API створене за допомогою GraphQL дозволяє створювати endpoint, який працює зі складними запитами та надає даним таку форму, яку потребує клієнтська сторона. Наприклад, якщо клієнту потрібні різні ресурси, то потрібно виконувати декілька запитів, щоб отримати усі потрібні дані(REST), але навіщо, якщо можна створити один запит на потрібні ресурси, а GraphQL їх збере і поверне у потрібному вигляді[5].

Приклад GraphQL запиту:

```
query getLocalnamespace($key: ID!) {
  locales(namespace: $key) {
    key
    ru
    en
  }
}
```

### 2.2.2 Вибір формату даних для обміну

Також важливим кроком є вибір формату даних для отримання результатів запиту з серверу.

Найпопулярнішими форматами для відносно невеликих розмірів даних є XML і JSON. XML - строгість форматування та простота перевірки.

Приклад формату XML:

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		17

```

<?xml version="1.0" encoding='UTF-8'?>
<users total="2">
  <user id="1">
    <username>alice</username>
    <email>alice@gmail.com</email>
  </user>
  <user id="2">
    <username>alex</username>
    <email>alex@gmail.com</email>
  </user>
</users>

```

Порівняно з другими форматами, XML виглядає дуже об'ємним. Кожен елемент даних повинен представитися детальною структурою, тому об'єм даних, що передається структурою виявляється малим.

Для аналізу даних такого формату також доводиться докладати чимало зусиль. Необхідно знати структури даних, а потім це все помістити в JS об'єкт. Тому це далеко не найкращий вибір формату даних для використання.

Таблиця 1 - Результати виміру продуктивності при використанні формату XML

Формат	Розмір	Час завантаження	Час аналізу	Загальний час
XML	582960 байт	999,4 мс	343,1 мс	1342,5 мс

JSON - формат, який є невеликим у розмірах та простим в аналізі форматом даних.

Нижче наводиться приклад списку користувачів:

```

[
  {"i": 1, "u": "alice", "e": "alice@gmail.com"},

```

```
{“i”: 2, “u”: “alex”, “e”: “alex@gmail.com”},
]
```

Як видно з прикладу список є масивом об'єктів, а кожен об'єкт - це користувач. Синтаксичний аналіз рядка очевидно буде швидшим і комфортнішим у використанні.

Таблиця 2 - Результати виміру продуктивності обробки списку JSON даних

Формат	Розмір	Час завантаження	Час аналізу	Загальний час
JSON	585790 байт	611,2 мс	37,2 мс	648,4 мс

Лише порівнюючи виміри можна вже виділити JSON, як більш вдалий вибір формату, а також аналізуючи те, що з XML працювати не так комфортно, зроблено однозначний вибір в сторону JSON.

### 2.2.3 Вибір бібліотеки для написання логіки

React - JavaScript бібліотека з відкритим первинним програмним кодом для розробки користувацьких інтерфейсів.

React розробляється та підтримується Facebook, Instagram і товариством окремих розробників та корпорацій.

React може використовуватись для розробки односторінкових і мобільних додатків. Його ціль - надати високу швидкість, простоту та масштабування.

Приклад застосування:

```
<div id="myReactApp"></div>
```

```
<script type="text/babel">
```

```
class Greeter extends React.Component {
  render() {
```

```

        return <h1>{this.props.greeting}</h1>
      }
    }
  }

ReactDOM.render(<Greeter      greeting="Hello      World!"      />,
  document.getElementById('myReactApp'));
</script>

```

Параметри передаються від батьківських компонент до дочірніх. Компоненти отримують параметри, як множину незмінних значень, тому компонента не може не може напряму змінювати параметри, але може викликати їх зміну через callback функції.

React використовує віртуальний DOM, створює кеш структуру в пам'яті, що дозволяє вирахувати зміну між попереднім та теперішнім станом інтерфейсу для оптимально оновлення DOM. Тому при зміні в конкретній компоненті, бібліотека це вичислює та оновлює лише потрібну компоненту.

JSX - це розширення синтаксису JavaScript, що дозволяє використовувати синтаксиси схожий до HTML синтаксису для опису структури інтерфейсу. Як правило, усі компоненти написані за допомогою HTML, але з можливістю використовувати JavaScript[1].

Кожна компонента має методи життєвого циклу, що дозволяє програмісту запускати код на різних стадіях життєвого циклу.

- constructor - конструктор, в якому ініціалізується компонента;
- componentWillMount - викликається перед рендером;
- render - основний метод життєвого циклу, визивається при зміні даних компоненти для оновлення даних в інтерфейсі;

- `componentDidMount` - викликається при першому рендері компоненти, як правило для отримання даних з віддаленого джерела;
- `componentDidUpdate` - викликається після оновлення компоненти;
- `componentWillUnmount` - викликається перед видаленням компоненти з DOM;
- `shouldComponentUpdate` - викликається при оновленні props або state, ця функція повертає true або false, що означає оновлювати чи ні компоненту;
- `componentWillUpdate` - викликається перед оновленням компоненти;
- `componentWillReceiveProps` - викликається при оновленні props;
- `componentDidCatch` - цикл в якому ловляться помилки;

## 2.2.4 Вибір фреймворку для написання стилів

Bootstrap - відкритий набір інструментів для створення сайтів і веб-додатків. Містить HTML- та CSS-шаблони оформлення для веб-форм, кнопок, міток та інших компонентів інтерфейсу.

Основні інструменти:

- сітки - задані розміри колонок, які можна одразу ж використовувати;
- шаблони - фіксовані або резинові шаблони документів;
- медіа - надає можливість управляти відображеннями та відео;
- таблиці - засоби оформлення таблиць, навіть надання сортування;

- форми - класи для оформлення форм і деяких подій, що з ними відбуваються;
- навігація - класи для оформлення панелей, вкладок, переходів між сторінками, меню і панелі інструментів;
- алерти - оформлення діалогових вікон, підказок та спливаючих вікон;

В проєкті використовується Bootstrap 4, це старий варіант, який насправді був повністю переписаний.

- перехід до використання SASS(до речі також присутній в проєкті);
- підтримка flexbox(новий спосіб розстановки блоків на сторінці);
- заміна основної одиниці виміру с px на rem;
- збільшений шрифт;

та багато іншого.

Переваги Bootstrap 4:

### **Зменшення кількості часу, що витрачається на розробку**

Традиційно використання фреймворків і бібліотек значно полегшує роботу розробникам і дозволяє розробляти проєкти швидше. Bootstrap дає готові рішення: їх використання дозволяє скоротити час, що витрачається на створення макета сайту.

### **Адаптивність**

Bootstrap дозволяє створювати адаптивні сайти. Дизайн сайту буде коректно відображатися на екранах пристроїв різних розмірів незалежно від їх діагоналі.

### **Кроссбраузерність**

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		22

Сайти, зроблені з використанням Bootstrap, будуть однаково відображатися у всіх сучасних браузерах.

### **Легкість у використанні і швидкість в освоєнні**

Bootstrap легко використовувати в розробці, в ньому легко розібратися.

А ті, хто ще не знайомий з Bootstrap, зможуть швидко навчитися працювати з цим інструментом. І справа не тільки в його простоті, але і у великій кількості уроків та інструкцій, які можна знайти на просторах мережі.

### **Зрозумілий код**

Bootstrap дозволяє писати якісний і зрозумілий код, який легко зрозуміє інший розробник. Це значно спрощує розробку в команді.

### **Єдність стилів**

Елементи Bootstrap виглядають гармонійно між собою і дозволяють створювати сторінки і сайти в єдиному стилі.[7]

## **2.2.5 Вибір баз даних.**

Серед усіх існуючих SQL баз даних була обрана PostgreSQL через ряд переваг над іншими існуючими відкритими базами даних.

- Модель даних - PostgreSQL об'єктно-реляційна база даних, що гарантує підтримку користувацьких об'єктів і їх поведінка, включаючи типи даних, функції, операції, домени та індекси.
- Структури і типи даних - існує обширний список типів даних, які підтримує PostgreSQL(uuid, грошові, геометричні, бінарні типи і т.д.) Основне - підтримка JSON.
- Мережеві адреси - PostgreSQL забезпечує зберігання різних типів мережевих адрес.



- Розміри даних - PostgreSQL може опрацьовувати великі об'єми даних.

Таблиця 3 - об'єми даних за характеристиками для PostgreSQL

Характеристика:	Значення:
Максимальний розмір БД	Необмежений
Максимальний розмір таблиці	32 TB
Максимальний розмір рядка	1.6 TB
Максимальний розмір поля	1 GB
Максимальна кількість рядків в таблиці	Необмежено
Максимальна кількість стовпчиків у таблиці	250-1600 в залежності від типу стовпчика

Максимальна кількість індексів в таблиці	Необмежено
--	------------

- Цілісність даних - PostgreSQL намагається відповідати стандарту ANSI-SQL:2008, відповідає вимогам ACID(атомарність, узгодженість, ізольованість і надійність) і відомий своєю посиальною і транзакційною цілісністю. Первинні ключі, що обмежують і каскадні зовнішні ключі, унікальні обмеження, обмеження NOT NULL, перевірочні обмеження та інші функції забезпечення цілісності даних дають впевненість, що тільки коректні дані будуть збережені.

## 2.3 Огляд обраних API для використання у розробці

### Twitch API v5

За допомогою даного API можна отримати інформацію про канали. Наприклад: назва каналу, кількість підписників, статус каналу, кількість переглядів і т.д. Уся інформація надається у форматі JSON. Для інтеграції даного сервісу у ваш додаток необхідно зареєструвати ваш додаток на Twitch dev portal та отримати посилку на яку потрібно буде посилати ваші створені graphql запити[3]. Приклад відповіді:

```
{
  "_total": 2,
  "users": [
    {
      "_id": "44322889",
      "bio": "Just a gamer playing games and chatting. :)",
      "created_at": "2013-06-03T19:12:02.580593Z",
```

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		25

```

    "display_name": "dallas",
    "name": "dallas",
    "type": "staff",
    "updated_at": "2017-02-09T16:32:06.784398Z"
  },
  {
    "_id": "129454141",
    "bio": null,
    "created_at": "2016-07-13T14:40:42.398257Z",
    "display_name": "dallasnchains",
    "name": "dallasnchains",
    "type": "user",
    "updated_at": "2017-02-04T14:32:38.626459Z"
  }
]
}

```

## 2.4 Вибір засобів розробки

WebStorm - середовище розробки від JetBrains, було обране для створення, редагування та налагодження додатку. Дане середовище для розробки не є безкоштовним, але студент може отримати ліцензію на рік, а також є безкоштовний пробний період в 30 днів. Середовище підтримується Windows, Linux та іншими ОС. Редактор містить вбудовану консоль, навігацію по додатку, зручне використання git, а також багато плагінів, які полегшують розробку, а також пришвидшують її.

Для автоматизації збірки додатку (мініфікації, тестів і т.д.) використовується Webpack.

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		26

Webpack - це статичний модульний збірник для додатків на JavaScript. Додатки, написані на JavaScript, постійно ускладнюються, тому хорошим вибором є використання бандлера(збірника). Подібні інструменти дозволяють розробникам упаковувати, компілювати і в цілому організовувати усі ресурси, необхідні для проекту. Можна використовувати не тільки сторонні бібліотек, але і власні файли. Подібна модульна система дозволяє досягти кращої організації проекту, так як виходить, що він розбитий на підмодулі. Webpack на цей момент один із найпотужніших подібних інструментів. Він має відкритий первинний програмний код і дозволяє вирішувати багато задач. Плюси:

- відмінно підходить для односторінкових додатків;
- може виконувати продвинутий розділ коду;

Мінуси:

- частина документації вже не актуальною;

Git - розподілена система керування версіями файлів. Основні можливості цього сховища:

- дублювання коду, або форкінг;
- запити на включення у проект змін чи злиття коду;
- можливість копіювати будь-який код з публічних репозиторіїв до власного для подальшої модифікації;
- можливість поділитися змінами із розробником оригінального коду та подати запит на включення цих змін до оригінального коду;
- спільне відстеження та коментування помилок;
- вікі-сторінки для проектів;
- керування задачами;
- підхід соціальної мережі, запроваджений для коду;

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		27

Docker - інструментарій для управління ізольованими Linux-контейнерами. Основні можливості Docker:

- Можливість розміщення в ізольованому оточенні різномірної начинки, що включає різні комбінації виконуваних файлів, бібліотек, файлів конфігурації, скриптів, файлів jar, gem, tar тощо.
- Підтримка роботи на будь-якому комп'ютері на базі архітектури x86\_64 з системою на базі ядра Linux, починаючи від ноутбуків, закінчуючи серверами та віртуальними машинами. Можливість роботи поверх немодифікованих сучасних ядер Linux (без накладення патчів) і в штатних оточеннях всіх великих дистрибутивів Linux, включаючи Fedora, RHEL, Ubuntu, Debian, SUSE, Gentoo і Arch.
- Використання легковагих контейнерів для ізоляції процесів від інших процесів і основної системи.
- Оскільки контейнери використовують свою власну самодостатню файлову систему, не важливо де, коли і в якому оточенні вони запускаються.
- Ізоляція на рівні файлової системи: кожен процес виконується у повністю окремій кореневій ФС.
- Ізоляція ресурсів: споживання системних ресурсів, таких як витрата пам'яті і навантаження на CPU, можуть обмежуватися окремо для кожного контейнера з використанням cgroups.
- Ізоляція на рівні мережі: кожен ізольований процес має доступ тільки до пов'язаного з контейнером мережевого простору імен,

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		28

включаючи віртуальний мережевий інтерфейс і прив'язані до нього IP-адреси.

- Коренева файлова система для контейнерів створюється з використанням механізму cory-on-write(окремо зберігаються тільки змінені і нові дані), що дозволяє прискорити розгортання, знижує витрату пам'яті і економить дисковий простір.
- Всі стандартні потоки (stdout/stderr) кожного виконуваного в контейнері процесу накопичуються і зберігаються у вигляді логу.
- Змінена файлова система одного контейнера може використовуватися як основа для формування нових базових образів і створення інших контейнерів, без необхідності оформлення шаблонів або ручного налаштування складу образів.
- Можливість використання інтерактивної командної оболонки: до стандартного вводу будь-якого контейнера може бути прив'язаний псевдо-tty для запуску shell.
- Підтримка використання різних систем зберігання, які можуть підключатися як плагіни. Серед підтримуваних драйверів зберігання заявлені aufs, device mapper (використовуються снапшоти LVM), vfs (на основі копіювання директорій) і Btrfs. Очікується поява драйверів для ZFS, Gluster і Ceph.
- Можливість створення контейнерів, що містять складні програмні стеки, через зв'язування між собою вже існуючих контейнерів, що містять складові частини формованого стека. Зв'язування здійснюється не через злиття вмісту, а через забезпечення взаємодії між контейнерами (створюється мережевий тунель).[23]

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		29

### 3. СТРУКТУРНО-АЛГОРИТМІЧНА ОРГАНІЗАЦІЯ

#### 3.1 Загальний опис структури та роботи файлів проекту

В даному проекті містяться кілька основних файлів, створених та написаних розробником, а також файли створені за допомогою коробки create-react-app, яка використовується для простого налагодження проекту. Але мова буде йти лише про власні файли.

**index.html** – кореневий файл проекту в якому знаходиться кореневий тег React додатку разом з прелоадером для додавання інтерактивності.

**index.scss** – файл для збереження стилів головної сторінки.

**styles.css** – ті ж самі стилі з index.scss, але скомпільовані у зрозумілий для браузера вид.

**src** – директорія в якій міститься уся логіка проекту, компоненти, стилі, роутинг, сервіси, конфіги і т.д.

**streamer.jsx** - компонента, яка містить всю інформацію про стрімера, його статистику, а також в цій компоненті виконується регулярний запит до API, для отримання нових даних.

**player.jsx** - компонента, яка містить iframe та кастомні контроли, дана компонента відображається лише у випадку, якщо канал даного стрімера онлайн.

**controls.jsx** - компонента, що містить кастомні контроли та методи управління плеєром.

**handleMouseMove()** - метод, що відображає контроли при активності курсору на елементі-плеєр.

**playerReady()** - метод, що перевіряє чи встановлений канал онлайн в даний момент часу.

**watchQuality()** - метод, що встановлює якість відображення у плеєрі.

**enterFullScreenMode()** - метод, для входу в повноекранний режим перегляду та налаштування додаткових функцій.

**exitFullScreenMode()** - метод, для виходу з повноекранного режиму перегляду та вимкнення додаткових функцій.

**handleVolumeChange()** - метод для зміни гучності.

**toggleMuted()** - метод, повного включення/вимкнення гучності.

**adminPanel** - директорія, яка містить настройки сторінки стрімера.

**streams.jsx** - компонента панелі адміна, де здійснюється привязка каналів, та вибір який відображається в даний момент.

**createStream.jsx** - компонента, що містить модальне вікно для створення каналу.

**getCreateStreamModal()** - метод для виклику модального вікна з формою створення нового каналу.

**settings.jxs** - деякі настройки конфігурацій, які дозволяють зробити вигляд сторінки дещо інакшим, наприклад змінити основні кольори, ввімкнути вимкнути чат в правому сайдбарі, встановити фото, опис, ім'я і т.д.

**linkChangeModal** - компонента, що вмістить модальне вікно створення записів на двох мовах.



**updateConfig()** - виклик мутації, яка змінить дані в БД згідно отриманих даних.

**getBooleanConfigForm()** - форма-чекбокс.

**getFooterConfig()** - форма, де можна додати соціальні мережі.

**\*.graphql** - усі файли даного розширення це файли які містять модель запиту до API, в яких вказується які конкретно дані треба повернути.

**layout.jsx** - компонента, яка містить футер, заголовок сторінки правий та лівий сайдбари, винесено в окремо компоненту, з метою розширення контенту на майбутнє, наприклад турніри з якоюсь статистикою матчів, але щоб каркас залишався однаковим.

**api.js** - містить query, endpoint, mutation для збереження конфігурацій, авторизації та каналів в бд.

**register()** - ендпоінт для реєстрації нового юзера.

**auth()** - ендпоінт для авторизації вже існуючого юзера.

**me()** - ендпоінт для отримання даних поточного користувача.

**streamer()** - ендпоінт для отримання даних і конфігурацій для відображення сторінки стрімера.

**pay()** - ендпоінт для проведення платежу до аккаунту стрімера.

**routes.js** - налаштування маршрутизації.

## 4. ОПИС ПРОГРАМНОГО ПРОДУКТУ

### 4.1 Принцип функціонування додатку.

При вході на сторінку стрімера та авторизації ми бачимо подібний контент, як на рисунку 4.1.

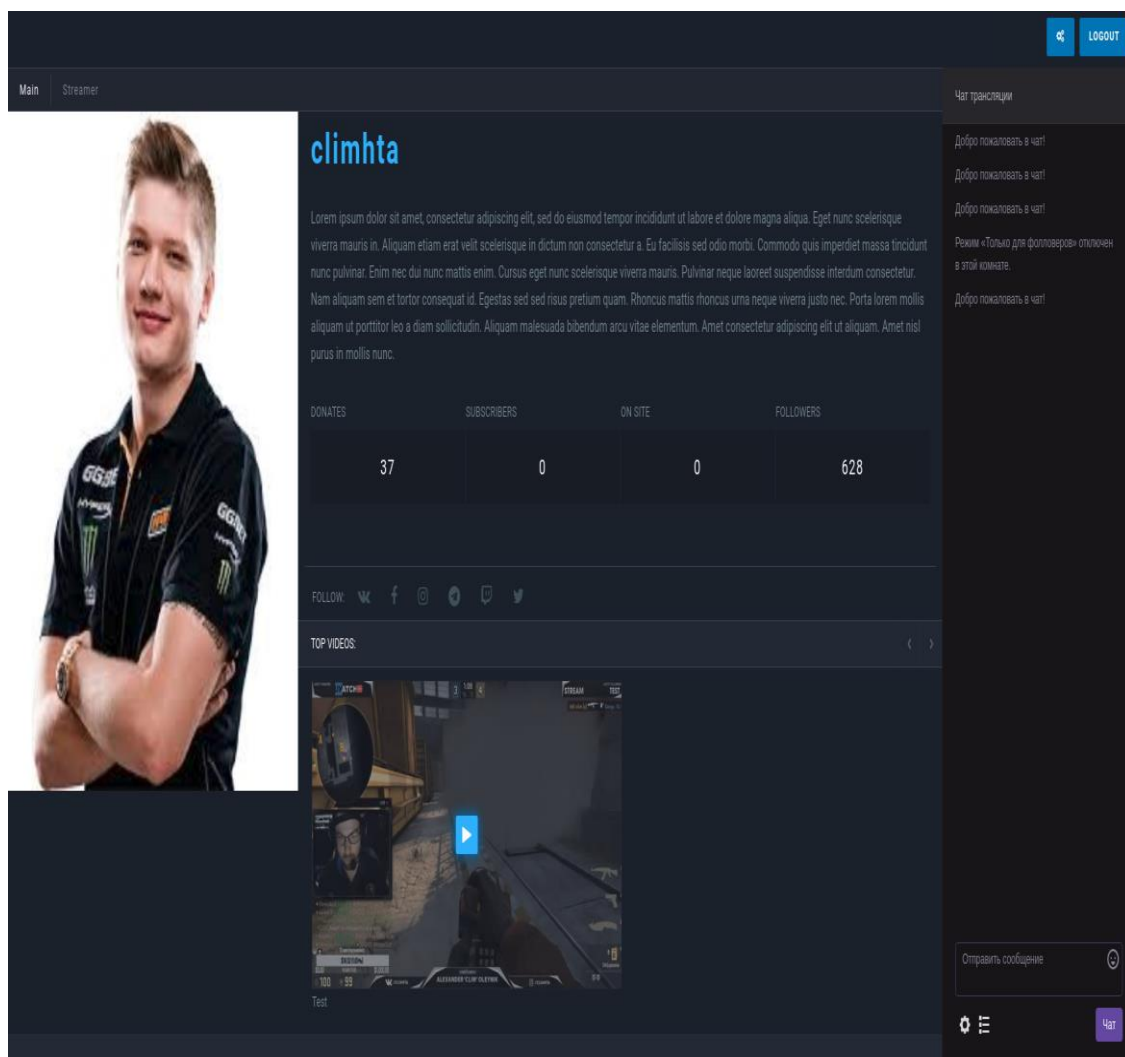


Рисунок 4.1 – Десктопна версія додатку

Також для прояву адаптивності веб-сторінки, можна переглянути мобільну версію (рисунок 4.2).



Рисунок 4.2 – Мобільна версія додатку

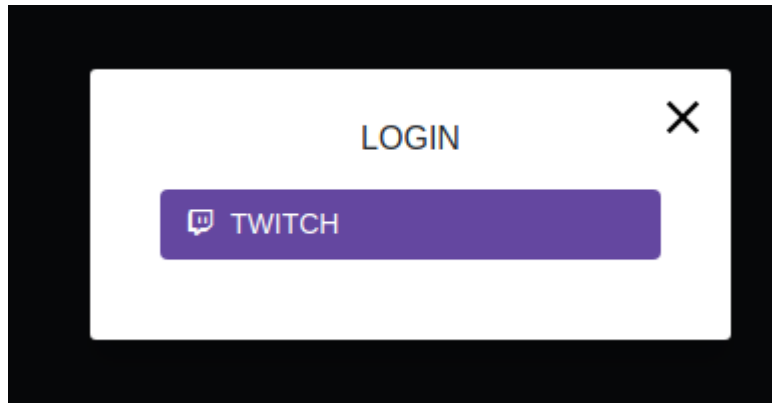


Рисунок 4.3 – Модальне вікно авторизації

Користувачеві надається можливість перейти до налаштувань власної сторінки. В хедері кнопка налаштування. При натисканні потрапляємо до адмін панелі і маємо можливість змінювати свою власну сторінку на власний розсуд, для цього надаються деякі конфігурації, які можна легко змінювати

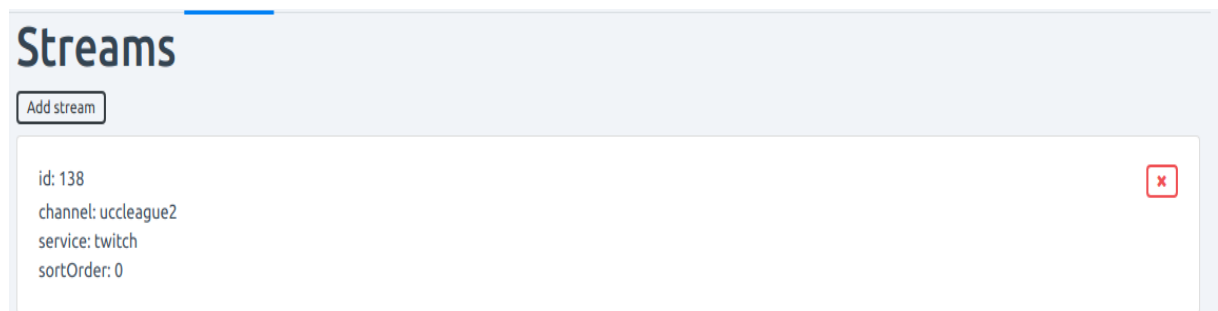


Рисунок 4.3 – Налаштування з додаванням стріму з twitch платформи

При натисканні на кнопку додати add stream викликається модальне вікно з формою для заповнення, щоб додати канал до свого аккаунту. Де потрібно вказати назву вашого каналу зареєстрованого на платформі Twitch.tv,

## Add new stream ✕

Stream link

Ex: food, but **not** <https://www.twitch.tv/food>

Select service

Twitch ▼

Save

Рисунок 4.4 – Форма додавання стріму до аккаунту

Одразу ж перевіряємо і на сторінці з'являється на самому початку трансляція цього каналу

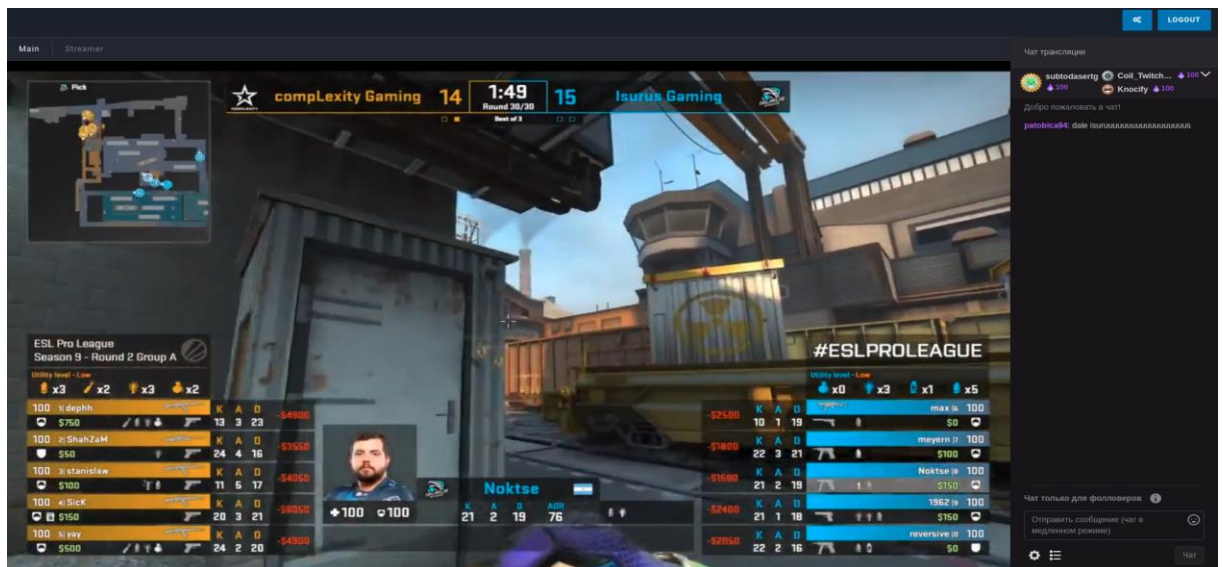


Рисунок 4.5 – Сторінка стрімера під час онлайн трансляції

і підключається чат з цієї ж трансляції.

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		36

Також користувач має можливість змінити кольори чи свої дані в налаштуваннях

Name: Clim  
Url: climhta  
Image: /static/41b026fc80.png

☒ Chat

Рисунок 4.6 – Основні налаштування

Name: Clim  
Url:    
Image: /static/41b026fc80.png

☒ Chat

Рисунок 4.7 – Основні налаштування з відкритою формою

Якщо вимкнути чат, сторінка залишається попередньою, але в правій частині екрану не відображається чат.

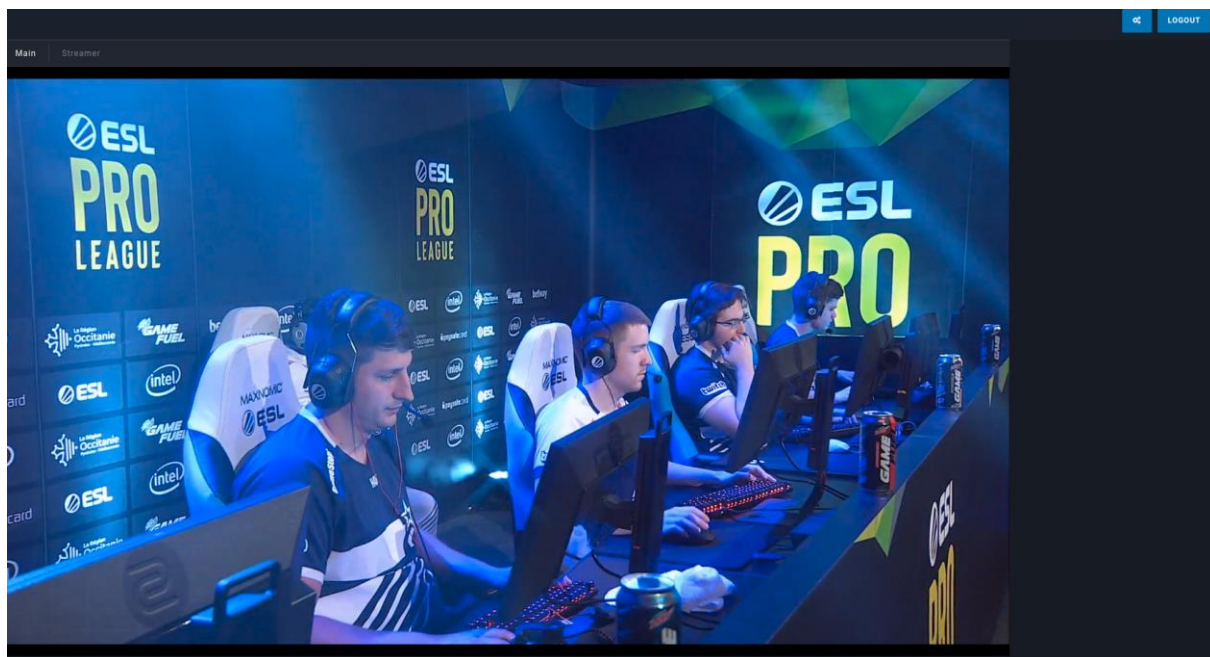


Рисунок 4.8 – Сторінка з вимкненими додатками в правому сайдбарі  
Також в налаштуваннях можна змінити опис та заголовок сторінки.

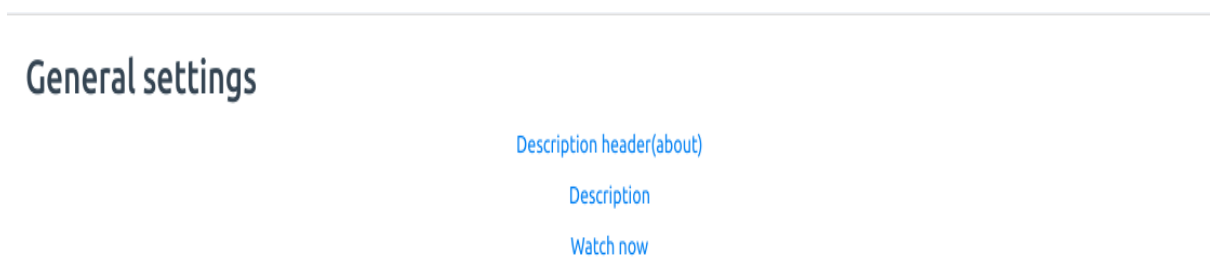


Рисунок 4.9 – Налаштування тексту в контенті сторінки  
Опис сторінки, заголовок та інші написи на сторінці заповнюються двома мовами, та відображаються залежно від вибору мови

#### EN translation

Lorem ipsum dolor sit amet, consectetur  
 adipiscing elit, sed do eiusmod tempor incididunt

#### RU translation

Lorem ipsum dolor sit amet, consectetur  
 adipiscing elit, sed do eiusmod tempor incididunt

#### Description

You can enter description here

Description is visible for admins only

#### Warning!

Locale will not be changed instantly

Save

You can submit by **Ctrl + Enter**

Рисунок 4.10 – Форма для заповнення контенту двома мовами

На сторінці стрімера можна побачити деякі характеристики, по типу підписники, кількість людей на трансляції на конкретний момент. Всі ці дані отримуються з API, кожних 5 хвилин отримуються свіжі дані по трансляції, які записуються в бд.



DONATES	SUBSCRIBERS	ON SITE	FOLLOWERS
0	0	1	1

Рисунок 4.11 – Відображення даних, доступних усім користувачам

Також можна увімкнути можливість донату, тоді внизу з'являться таби для переключення між чатом та донатом в правому сайдбарі.

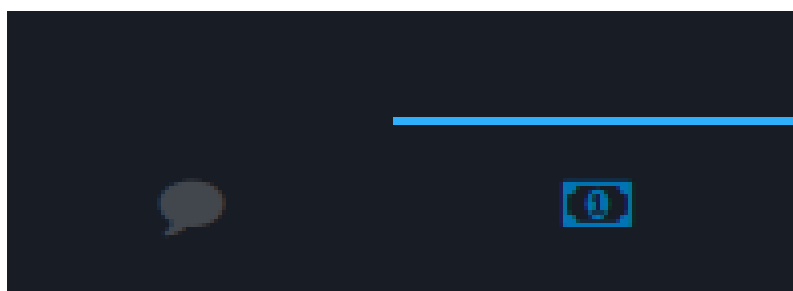


Рисунок 4.12 – Перемикач вкладок на боковій панелі

При переключенні отримаємо іншу компоненту в правому сайдбарі, яка дозволяє провести платіж стрімеру, містить 4 поля для введення:

1. кількість одиниць валюти;
2. валюта;
3. повідомлення, яке отримує стрімер на трансляції;
4. ім'я відправника, якщо порожнє то прийде анонім, мінімум 3 символи;

**DONATION AMOUNT\***

100

UAH ▾

\*Service fee is not included

\*Currency will be converted to streamer currency

**DONATION MESSAGE**

Text

255

**DONATION USERNAME**

Yakovenchik

21

DONATE

Рисунок 4.13 – Вкладка платежів на боковій панелі

Присутня можливість вибору валюти

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		41

**DONATION AMOUNT\***

100

UAH ^

USD

\*Service fee is not included

\*Currency will be converted to streamer currency

Рисунок 4.14 – Вибір валют

Демонстрація валідації введених полів. Якщо ввести менше трьох символів, кнопка платежу неактивна і поле підсвічується червоним.

**DONATION USERNAME**

12 30

Рисунок 4.15 – Невалідні дані

**DONATION USERNAME**

123 29

Рисунок 4.15 – Валідні дані

Також можна помітити, на полях для введення лічильники, які вказують скільки ще можна ввести символів, поля для введення налаштовані так, що не дозволяють вводити більше символів, ніж вказано

на лічильнику при пустому полі. Після введення усіх валідних даних, якщо натиснути на кнопку надіслання транзакції, буде вимагатися ще одна перевірка у виді рекапчі.

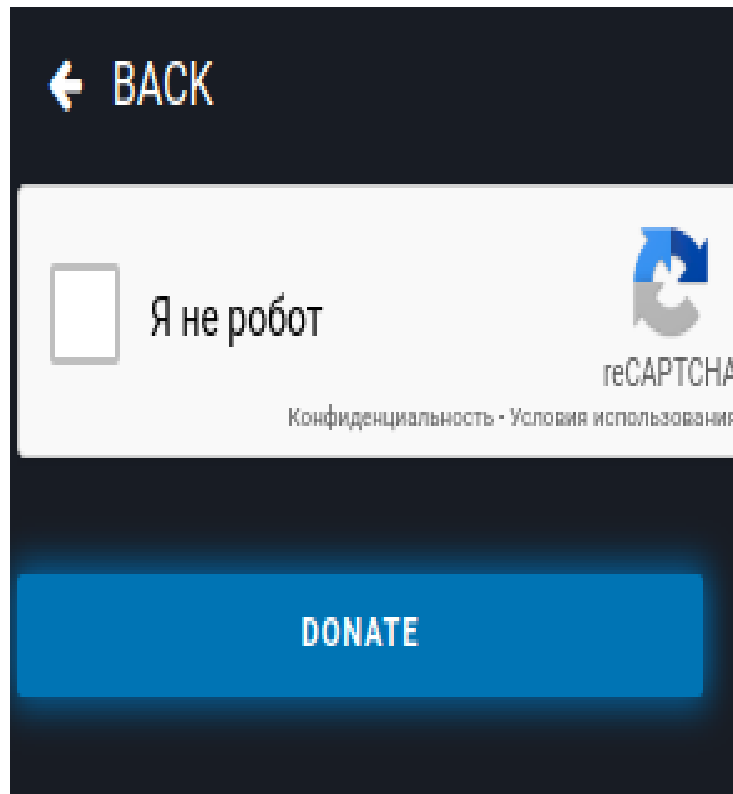


Рисунок 4.15 – Перевірка для відправлення платежу

В налаштуваннях сторінки, також можна побачити графіки по даним, які отримуються з початку додавання каналу до своєї сторінки за весь період.

UNIQUE VISITORS

12,537

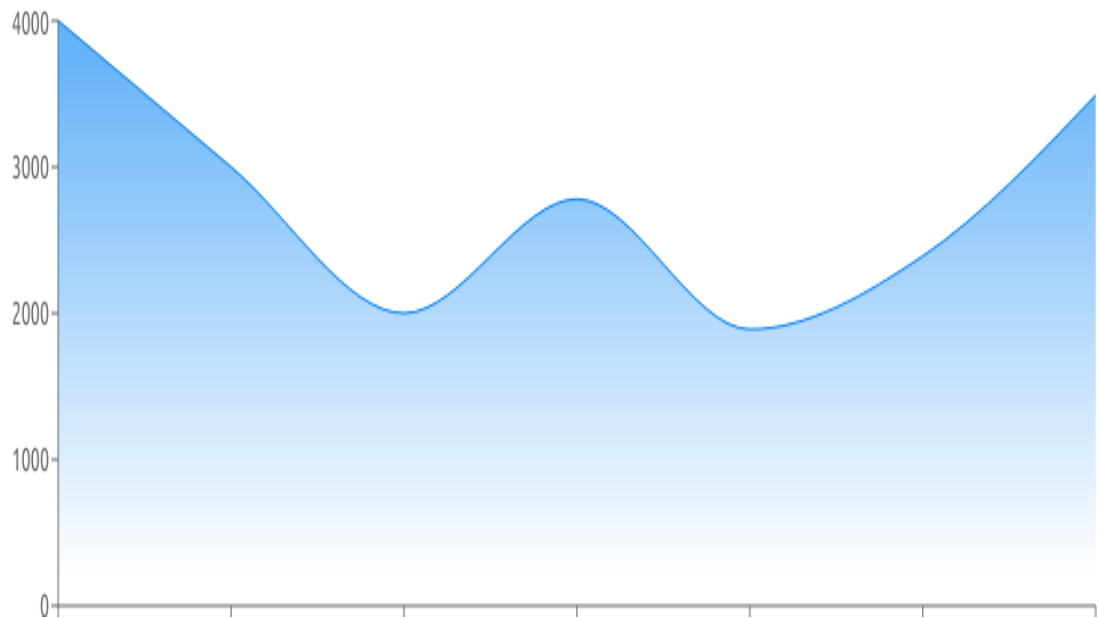


Рисунок 4.16 – Графік підписників

Також можна побачити всю інформацію по платежам проведеним через платформу:

- кількість платежів;
- скільки постуило грошей у валюті вказаній для стрімера, усі надходження конвертуються в одну валюту;
- кількість нових підписників;
- скільки грошей отримано за день;

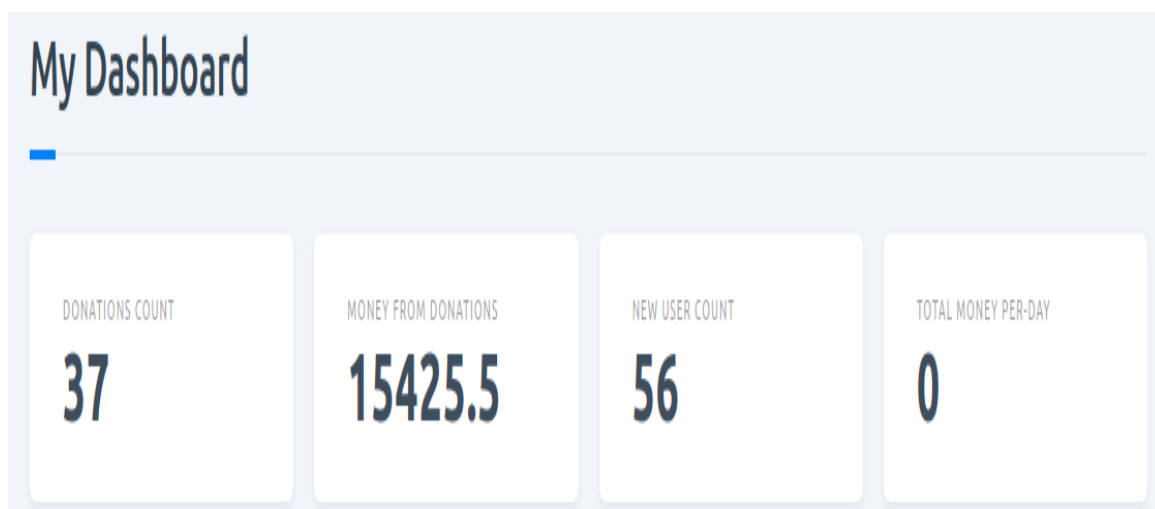


Рисунок 4.17 – Статистика по аккаунту в адмін панелі

Також можна отримати історію останніх подій, які відбулись з конкретним аккаунтом на платформі, де буде вказано коли відбулась подія, тип події, та її статус, або ім'я власника події, тощо.

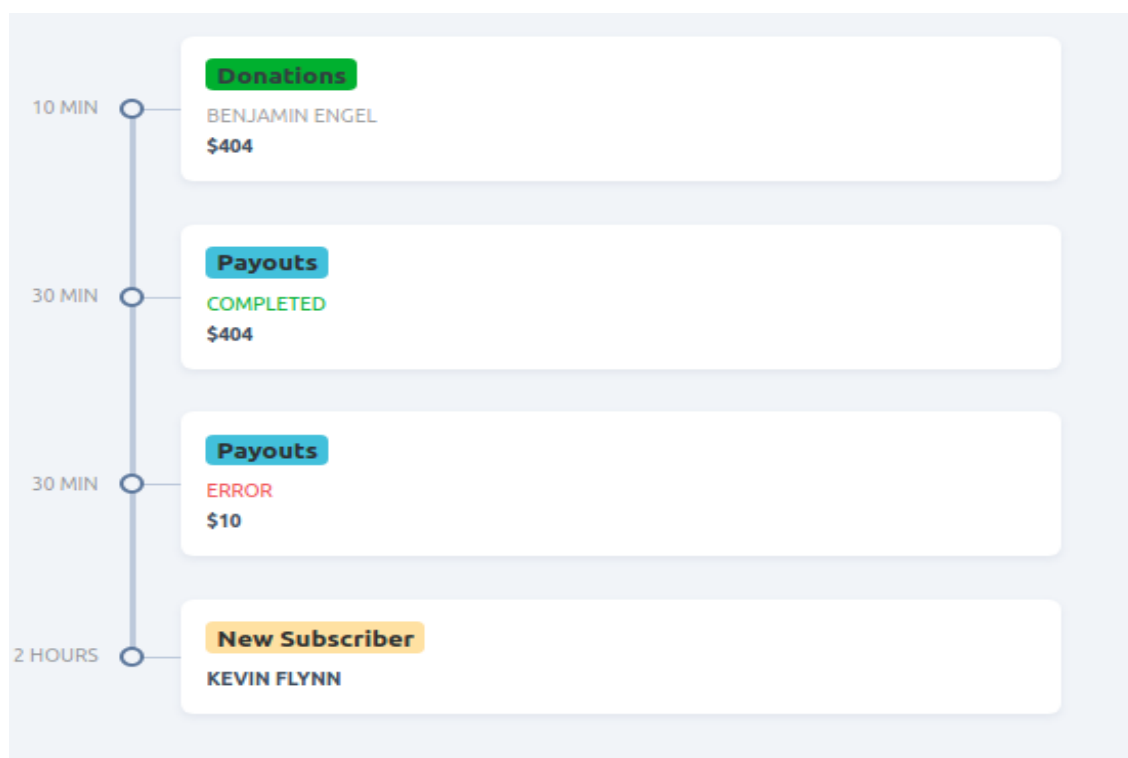


Рисунок 4.18 – Історія подій в адмін панелі

Також історію усіх надходжень з іменами відправників, кількістю грошей та дату надходження. Відображення статусів по транзакції залежно від її успішності.

DONATOR NAME	CURRENCY	AMOUNT	DATE
ECOMMPAY	UAH	120	16-05-2019
ECOMMPAY	UAH	213	16-05-2019
asd	UAH	1	16-05-2019
makentaer	UAH	1111	16-05-2019
makentaer	UAH	101	16-05-2019
Red	UAH	1	16-05-2019
makentaer	UAH	1111	16-05-2019
Red	UAH	1	16-05-2019
m2kentaer	UAH	333	16-05-2019
m2kentaer	UAH	101	16-05-2019
Anonymous	UAH	111	16-05-2019
Anonymous	UAH	1000	16-05-2019
Red	UAH	1.5	16-05-2019
Anonymous	UAH	100	15-05-2019
fullstackoferflow dev	UAH	123	15-05-2019
fullstackoferflow dev	UAH	1	15-05-2019
КАПЕЦ	UAH	1010	15-05-2019
fullstackoferflow dev	UAH	123	15-05-2019
Anonymous	UAH	1231	15-05-2019

Рисунок 4.19 – Історія платежів

## ВИСНОВКИ

Під час написання дипломного проекту було розглянуто питання доцільності використання готових програмних інтерфейсів у розробці власних проектів, додатків, веб-сайтів, тощо. Було розроблено міні-платформу для стрімерів і використано чимало сучасних бібліотек та розробок. В процесі виконання було виявлено, що гарно написані програмні інтерфейси(API) дуже легко інтегрувати у власні системи, а також полегшують та пришвидшують розробку. Використання GraphQL API у проектах, також є гарним впровадженням, адже використання даної технології допомагає уникнути великої кількості запитів, що полегшує розробку та зменшує навантаження на БД. Інтеграція елементів інших платформ та відомих проектів через <iframe>, дозволяє уникнути затрат на підтримку даних елементів, а також звільняє від їх розробки. Також використання контейнерів в docker є дуже доцільним, що допомагає швидко і легко налаштувати базу даних на потрібний порт, та згодом імпортувати та експортувати дані. Загалом можна зробити висновок, що використання сучасних технологій є явно доцільним, тому потрібно вивчати усі нововведення та використовувати їх у власній розробці.

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		47



## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. React - a JavaScript Library for building user interfaces. – Режим доступу: <https://reactjs.org/>
2. What's an API and Why is it Important? [Електронний ресурс] / CNET Networks, Inc. 2010 - Режим доступу: <https://www.zdnet.com/article/whats-an-api-and-why-is-it-important/>
3. The future of live entertainment is here – Режим доступу: <https://dev.twitch.tv/why-twitch>
4. The Importance of APIs for Business [Електронний ресурс] / Moeller D.F., 2016 – Режим доступу: <https://www.glowtouch.com/importance-apis-business/>
5. GraphQL - a query language for your API – Режим доступу: <https://graphql.org/>
6. API від Watson та що ці інструменти можуть дати вашому сервісу чи додатку [Електронний ресурс] / Мельник А.М. - Режим доступу: <https://habr.com/company/ibm/blog/331206/>
7. Bootstrap – Режим доступу: <https://getbootstrap.com/docs/4.3/getting-started/introduction/>
8. 15 APIs developers need to know [Електронний ресурс] / DuVander A., 2015 – Режим доступу: <https://www.creativebloq.com/web-design/apis-developers-need-know-121518469>
9. Dropbox Tips & tricks [Електронний ресурс] / Tendre F. G., 2018 – Режим доступу: <https://blogs.dropbox.com/business/category/tips-tricks/>
10. HTML <a> href Attribute [Електронний ресурс] / W3Schools - The World's Largest Web Development Portal – Режим доступу: [https://www.w3schools.com/tags/att\\_a\\_href.asp](https://www.w3schools.com/tags/att_a_href.asp)

11. Enterprise Container Platform for High-Velocity Innovation—  
Режим доступу:<https://www.docker.com/>
12. CSS властивості [Електронний ресурс] / Український веб-  
довідник – Режим доступу: <https://css.in.ua/css/properties>
13. Фрейн Б., HTML5 и CSS3. Разработка сайтов для любых  
браузеров и устройств. — СПб.: Питер, 2014. — 151-182 ст.
14. JavaScript [Електронний ресурс] / Mozilla Developers Network –  
Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/>
15. [MacCaw](#) А.В., The little book on CoffeeScript, — San Francisco,  
2012 — 53-73 ст.
16. Nicholas C. Zakas, Javascript Optimization, 1005 Gravenstein  
Highway North, 2012 – 112-124 ст.
17. Data Format Test [Електронний ресурс] / Ross Harmes, Tristan  
Roy – Режим доступу: <http://techfoolery.com/formats/>
18. W3C Markup Validation Service [Електронний ресурс] / World  
Wide Web Consortium – Режим доступу: <https://validator.w3.org>
19. Docker – Режим доступу:  
<https://www.docker.com/products/docker-enterprise>
23. Understanding The Full Impact Of Web Design On SEO, Branding,  
And More [Електронний ресурс] – Drew Hendricks, 2015 – Режим  
доступу: <https://goo.gl/5tG4qq>
24. Wikipedia – Режим доступу: <https://uk.wikipedia.org/>
25. Sass – as a professional grade CSS extension language [Електронний  
ресурс] / Hampton Catlin, Natalie Weizenbaum, Chriss Eppstein – Режим  
доступу: <http://sass-lang.com/>
26. Automate and enhance your workflow [Електронний ресурс] / Fractal  
Innovations – Режим доступу: <https://gulpjs.com/>

27. Зачем нужны вендорные префиксы [Электронный ресурс] / Игорь Муратов, Валерий Лобанов, 2015 – Режим доступа: <https://vaden-pro.ru/blog/css/vendorsnye-prefiksy>

28. Pro Git [Электронный ресурс] / Scott Chacon, Ben Straub – Режим доступа: <https://git-scm.com/book/en/v2>

29. Twitch Documentation – Режим доступа: <https://twitch.tv>

30. YouTube – Режим доступа: <https://youtube.com>

31. CyberGame – Режим доступа: <https://cybergame.tv>

					ІАЛЦ.045440.004 ПЗ	Арк.
Изм.	Лист	№ докум.	Підпис	Дата		50